

PCT

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference AJR/40161	FOR FURTHER ACTION see Notification of Transmittal of International Search Report (Form PCT/ISA/220) as well as, where applicable, item 5 below.	
International application No. PCT/GB 99/ 03711	International filing date (day/month/year) 08/11/1999	(Earliest) Priority Date (day/month/year) 06/11/1998
Applicant IMAGINATION TECHNOLOGIES LIMITED et al.		

This International Search Report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This International Search Report consists of a total of 3 sheets.

☒ It is also accompanied by a copy of each prior art document cited in this report.

1. Basis of the report

a. With regard to the **language**, the international search was carried out on the basis of the international application in the language in which it was filed, unless otherwise indicated under this item.

☐ the international search was carried out on the basis of a translation of the international application furnished to this Authority (Rule 23.1(b)).

b. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international search was carried out on the basis of the sequence listing :

☐ contained in the international application in written form.

☐ filed together with the international application in computer readable form.

☐ furnished subsequently to this Authority in written form.

☐ furnished subsequently to this Authority in computer readable form.

☐ the statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.

☐ the statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished

2. ☐ **Certain claims were found unsearchable** (See Box I).

3. ☐ **Unity of invention is lacking** (see Box II).

4. With regard to the **title**,

☒ the text is approved as submitted by the applicant.

☐ the text has been established by this Authority to read as follows:

5. With regard to the **abstract**,

☐ the text is approved as submitted by the applicant.

☒ the text has been established, according to Rule 38.2(b), by this Authority as it appears in Box III. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

6. The figure of the **drawings** to be published with the abstract is Figure No.

☐ as suggested by the applicant.

☒ because the applicant failed to suggest a figure.

☐ because this figure better characterizes the invention.

☒ None of the figures.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/GB 99/03711

Box III TEXT OF THE ABSTRACT (Continuation of item 5 of the first sheet)

A texturing system for use in a three-dimensional imaging system comprises a memory (22) for storing mip-map data for use in texturing an image. A controller (24) retrieves from the memory the mip-map data required and this data is stored in a cache (30). A lower-level mip-map generator (36) generates portions of the mip-map which is next below, in the hierarchical series, the mip-map of which portions are held in the cache. A trilinear interpolator (34) interpolates one output texel from input texels from the two mip-map levels. The texture data is represented by compressed codes. The lower-level mip-map generator (36) interpolates on the compressed code values.

INTERNATIONAL SEARCH REPORT

International Application No

PCT/GB 99/03711

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06T15/20

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
------------	--	-----------------------

A	EP 0 810 553 A (SEIKO EPSON CORP) 3 December 1997 (1997-12-03) ----	
A	WO 97 41534 A (3DFX INTERACTIVE INC) 6 November 1997 (1997-11-06) ----	
A	US 5 651 104 A (COSMAN MICHAEL A) 22 July 1997 (1997-07-22) -----	

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

3 February 2000

Date of mailing of the international search report

10/02/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Burgaud, C

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/GB 99/03711

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
EP 0810553	A	03-12-1997	US 5886705 A	23-03-1999
			JP 10105723 A	24-04-1998
WO 9741534	A	06-11-1997	US 5831624 A	03-11-1998
			AU 2732797 A	19-11-1997
US 5651104	A	22-07-1997	NONE	

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

(PCT Article 36 and Rule 70)

Applicant's or agent's file reference AJR/40161	FOR FURTHER ACTION See Notification of Transmittal of International Preliminary Examination Report (Form PCT/IPEA/416)	
International application No. PCT/GB99/03711	International filing date (day/month/year) 08/11/1999	Priority date (day/month/year) 06/11/1998
International Patent Classification (IPC) or national classification and IPC G06T15/20		
Applicant IMAGINATION TECHNOLOGIES LIMITED et al.		

1. This international preliminary examination report has been prepared by this International Preliminary Examining Authority and is transmitted to the applicant according to Article 36.



2. This REPORT consists of a total of 9 sheets, including this cover sheet.

- ☒ This report is also accompanied by ANNEXES, i.e. sheets of the description, claims and/or drawings which have been amended and are the basis for this report and/or sheets containing rectifications made before this Authority (see Rule 70.16 and Section 607 of the Administrative Instructions under the PCT).

These annexes consist of a total of 4 sheets.

3. This report contains indications relating to the following items:

- I ☒ Basis of the report
- II ☐ Priority
- III ☒ Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
- IV ☒ Lack of unity of invention
- V ☒ Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
- VI ☐ Certain documents cited
- VII ☒ Certain defects in the international application
- VIII ☒ Certain observations on the international application

Date of submission of the demand 06/06/2000	Date of completion of this report 02.02.2001
Name and mailing address of the international preliminary examining authority:  European Patent Office D-80298 Munich Tel. +49 89 2399 - 0 Tx: 523656 epmu d Fax: +49 89 2399 - 4465	Authorized officer Knoepfle, R Telephone No. +49 89 2399 2659 

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No. PCT/GB99/03711

I. Basis of the report

1. This report has been drawn on the basis of *(substitute sheets which have been furnished to the receiving Office in response to an invitation under Article 14 are referred to in this report as "originally filed" and are not annexed to the report since they do not contain amendments (Rules 70.16 and 70.17).):*

Description, pages:

1-24 as originally filed

Claims, No.:

1-9 as received on 28/11/2000 with letter of 24/11/2000

Drawings, sheets:

1/9-9/9 as originally filed

2. With regard to the **language**, all the elements marked above were available or furnished to this Authority in the language in which the international application was filed, unless otherwise indicated under this item.

These elements were available or furnished to this Authority in the following language: , which is:

- ☐ the language of a translation furnished for the purposes of the international search (under Rule 23.1(b)).
- ☐ the language of publication of the international application (under Rule 48.3(b)).
- ☐ the language of a translation furnished for the purposes of international preliminary examination (under Rule 55.2 and/or 55.3).

3. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international preliminary examination was carried out on the basis of the sequence listing:

- ☐ contained in the international application in written form.
- ☐ filed together with the international application in computer readable form.
- ☐ furnished subsequently to this Authority in written form.
- ☐ furnished subsequently to this Authority in computer readable form.
- ☐ The statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.
- ☐ The statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished.

4. The amendments have resulted in the cancellation of:

- ☐ the description, pages:
- ☐ the claims, Nos.:

**INTERNATIONAL PRELIMINARY
EXAMINATION REPORT**

International application No. PCT/GB99/03711

☐ the drawings, sheets:

5. ☐ This report has been established as if (some of) the amendments had not been made, since they have been considered to go beyond the disclosure as filed (Rule 70.2(c)):

(Any replacement sheet containing such amendments must be referred to under item 1 and annexed to this report.)

6. Additional observations, if necessary:

III. Non-establishment of opinion with regard to novelty, inventive step and industrial applicability

1. The questions whether the claimed invention appears to be novel, to involve an inventive step (to be non-obvious), or to be industrially applicable have not been examined in respect of:

☐ the entire international application.

☒ claims Nos. 8, 9 .

because:

☐ the said international application, or the said claims Nos. relate to the following subject matter which does not require an international preliminary examination (*specify*):

☒ the description, claims or drawings (*indicate particular elements below*) or said claims Nos. 8, 9 are so unclear that no meaningful opinion could be formed (*specify*):
see separate sheet

☐ the claims, or said claims Nos. are so inadequately supported by the description that no meaningful opinion could be formed.

☐ no international search report has been established for the said claims Nos. .

2. A meaningful international preliminary examination report cannot be carried out due to the failure of the nucleotide and/or amino acid sequence listing to comply with the standard provided for in Annex C of the Administrative Instructions:

☐ the written form has not been furnished or does not comply with the standard.

☐ the computer readable form has not been furnished or does not comply with the standard.

IV. Lack of unity of invention

1. In response to the invitation to restrict or pay additional fees the applicant has:

☐ restricted the claims.

INTERNATIONAL PRELIMINARY EXAMINATION REPORT

International application No. PCT/GB99/03711

- ☐ paid additional fees.
- ☐ paid additional fees under protest.
- ☐ neither restricted nor paid additional fees.
- 2. ☒ This Authority found that the requirement of unity of invention is not complied and chose, according to Rule 68.1, not to invite the applicant to restrict or pay additional fees.
- 3. This Authority considers that the requirement of unity of invention in accordance with Rules 13.1, 13.2 and 13.3 is
 - ☐ complied with.
 - ☒ not complied with for the following reasons:
see separate sheet
- 4. Consequently, the following parts of the international application were the subject of international preliminary examination in establishing this report:
 - ☐ all parts.
 - ☒ the parts relating to claims Nos. 1 - 7.

V. Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

1. Statement

Novelty (N)	Yes:	Claims 1 - 7
	No:	Claims
Inventive step (IS)	Yes:	Claims 1 - 7
	No:	Claims
Industrial applicability (IA)	Yes:	Claims 1 - 7
	No:	Claims

2. Citations and explanations **see separate sheet**

VII. Certain defects in the international application

The following defects in the form or contents of the international application have been noted:
see separate sheet

VIII. Certain observations on the international application

The following observations on the clarity of the claims, description, and drawings or on the question whether the claims are fully supported by the description, are made:

**INTERNATIONAL PRELIMINARY
EXAMINATION REPORT**

International application No. PCT/GB99/03711

see separate sheet

Re Item III

Non-establishment of opinion with regard to novelty, inventive step and industrial applicability

- 1). Claims 8 and 9 do not meet the requirements of Article 6 PCT in that the matter for which protection is sought is not clearly defined.
- 2). In lines 10 to 25 of page 22 of the description the Applicant writes that the normal way of producing actual colour values is to start with principal compressed colour values and to determine actual colour values by weighting and averaging selected principal colour values in the frame of the accuracy of the system which may require the complete bit-width.
- 3). The Applicant has, however, detected that it is convenient first to determine actual colour values in compressed form and then to convert the compressed colour values thus obtained to the nearest colour value (page 22, lines 16 to 25).
- 4). Said claims 8 and 9, however, defining in their characterising part that
- the interpolation is effected by (the interpolating means) using compressed code values,
define the subject-matter in terms of the result to be achieved which merely amounts to a statement of the underlying problem.
- 5). In order to meet the requirements of Article 6 PCT the technical features necessary for achieving this result should be added; in the present case

- index look-up table means (92) which provide indices of the principal colours;
- mip-map filters (66) which perform actual compressed weighted average values;
- codebook lookup means (94) where the filtered indices are used to find the corresponding codebook portion of an actual colour.

Re Item IV

Lack of unity of invention

6). The present application is directed to

- a texturing system for use in a three-dimensional imaging system (claim 1);
- a method of texturing for use in three-dimensional imaging (claim 7);
- an apparatus for generating text data for use in texturing an image (claim 8): and to
- a method of generating texture data for use in texturing an image (claim 9).

7). After clarification of claims 8 and 9 in point (5) two separate inventions appear to exist in the present application, namely one comprising claims 1 to 7 and the other comprising claims 8 and 9.

Since both inventions

- evidently start from different prior art (the first from Figure 4 of the present application; the second from the prior art cited in point (2);
- solve different technical problems (the first cited in point (11): the second cited in point (3) and
- provide different technical features,

the two inventions appear not to be so linked as to form a single general inventive concept (Rule 13.1 PCT).

Re Item V

Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

8) Relating to claims 1 to 7 reference is made to the following document:

D1: WO-A-9741534.

9). A method of texturing [and a texturing system], for use in three-dimensional imaging comprising the features
[memory means for] storing mip-map data:

[input means for] receiving input data indicating the type of mip-map data;
[control means for] retrieving from memory the mip-map data required in accordance with the input means;
[cache means for] storing portions of mip-map data retrieved from memory; and
[trilinear interpolator means for] interpolating an output texel from mip-map data stored in the cache
is known from the state of the art (present description page 16, line 14 to page 17, line 21).

10). Starting from said prior art it appears to be the problem of the present invention to reduce the bandwidth expressed in Bytes/s to an acceptable level (page 16, lines 26/27).

11). This problem is solved in independent claims 1 and 7 by
- [lower-level mip-map generator means for] generating in real time portions of the mip-map data next below in the hierarchal series of mip-map data of which portions are held in the cache means; and by
- [the trilinear interpolator means] interpolating an output texel from the mip-map level data stored in the cache means and the mip-map data next below in the hierarchal series generated in real time.

In this way the instantaneously generated lower bit-map data portion is used with the mip-map data read from the cache means for trilinear interpolation and is then discarded.

Reading only the " upper " mip-map data and computing only the " lower " mip-map data that is needed, have memory bandwidth and storage advantages.

12). Therefore independent claims 1 and 7 meet the requirements of Articles 33(2) and (3) PCT.

13). Dependent claims 2 - 6 specify advantageous embodiments of the system according to claim 1 and therefore also meet the requirements of Articles 33(2) and (3) PCT.

Re Item VII

Certain defects in the international application

14). Contrary to the requirements of Rule 5.1(a)(ii) PCT, the relevant background art disclosed in document D1 is not mentioned in the description, nor is this document identified therein.

15). Especially the features of the claims of the second invention are not provided with reference signs placed in parentheses (Rule 6.2(b) PCT).

Re Item VIII

Certain observations on the international application

16). Claims 8 and 9 define in their precharacterising part

- (means for) representing texture data by arbitrary compressed codes and
- (interpolating means) for interpolating an output texel from a plurality of input texels

and in their characterising part that

- the interpolation is effected by (the interpolating means) using compressed code values.

17). In contrast to independent claims 1 and 7 no mip-maps are introduced which can be manipulated on different levels of decreasing resolution. Therefore it cannot be understood how the different means cooperate since all means evidently handle data on different levels.

- 25 -

CLAIMS

1. A texturing system for use in a three-dimensional imaging system, and comprising:

memory means (22) for storing mip-map data for use in
5 texturing an image, the mip-map data comprising a hierarchical series of mip-maps of different levels of decreasing resolution;

input means (26) for receiving input data indicating the type of mip-map data required and the level of the
10 mip-map or mip-maps from which the data is to be taken;

control means (24) coupled to the input means and to the memory means for retrieving from memory the mip-map data required in accordance with the input data; and

cache means (30;60) coupled to the control means for
15 storing portions of mip-map data retrieved from memory and relating to a selected mip-map level; and

trilinear interpolator means (34;64) coupled to the cache means to receive mip-map data from one level of mip-map and to interpolate an output texel from input
20 texels from two mip-map levels;

characterised by:

lower-level mip-map generator means (36;66) coupled between the cache means and the trilinear interpolator means for generating portions of the mip-map next below in
25 the hierarchical series the mip-map of which portions are held in the cache means.

2. A texturing system according to claim 1, including decompression means (32;62) for decompressing mip-map data, the mip-map data being stored in the memory means in
30 compressed form.

3. A texturing system according to claim 2, in which the decompression means (32;62) is connected with its input

- 26 -

coupled to the cache means and its output coupled both to the trilinear interpolator means and to the lower-level mip-map generator means.

4. A texturing system according to claim 2 or 3, in
5 which there are four cache means (60) and four decompression means (62) arranged in parallel.

5. A texturing system according to claim 2, 3 or 4,
further comprising allocating means (82) between the cache
means (60) and the decompression means (62), for
10 allocating the outputs of different caches to selected ones of the decompression means.

6. A texturing system according to any preceding claim,
in which the lower-level mip-map generator means comprises
four interpolators which operate on 16 texels from the
15 mip-map held in the cache means to provide four texels as from the next lower mip-map.

7. A method of texturing for use in three-dimensional
imaging, and comprising the steps of:
storing in memory mip-map data for use in texturing
20 an image, the mip-map data comprising a hierarchical series of mip-maps of different levels of decreasing resolution;
receiving input data indicating the type of mip-map
data required and the level of the mip-map or mip-maps
25 from which the data is to be taken;
retrieving from memory the mip-map data required in accordance with the input data; and
storing in a cache portions of mip-map data retrieved from memory and relating to a selected mip-map level; and

- 27 -

receiving mip-map data from one level of mip-map and interpolating an output texel from input texels from two mip-map levels;

characterised by:

5 generating in real-time portions of the mip-map next below in the hierarchical series the mip-map of which portions are held in the cache means.

8. Apparatus for generating texture data for use in texturing an image, comprising the steps of:

10 means for representing texture data by arbitrary compressed codes, in which selected compressed code values define principal colors and other compressed code values define colors which can be formed by selected weighted averages of principal colors, the corresponding code values also being weighted averages of the code values of the selected principal colors; and

interpolating means for interpolating an output texel from a plurality of input texels;

characterised in that:

20 the interpolating means effects the interpolation using compressed code values.

9. A method of generating texture data for use in texturing an image, comprising the steps of:

25 representing texture data by arbitrary compressed codes, in which selected compressed code values define principal colors and other compressed code values define colors which can be formed by selected weighted averages of principal colors, the corresponding code values also being weighted averages of the code values of the selected principal colors;

30 interpolating an output texel from a plurality of input texels;

characterised in that:

- 28 -

the interpolating step is effected using compressed code values.

1 / 9

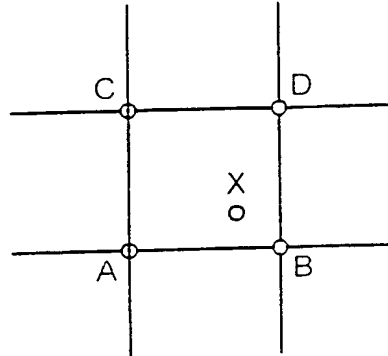


FIG. 1

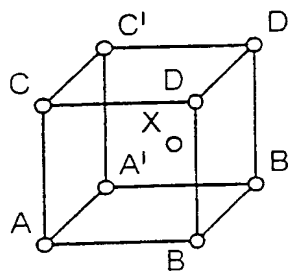


FIG. 2

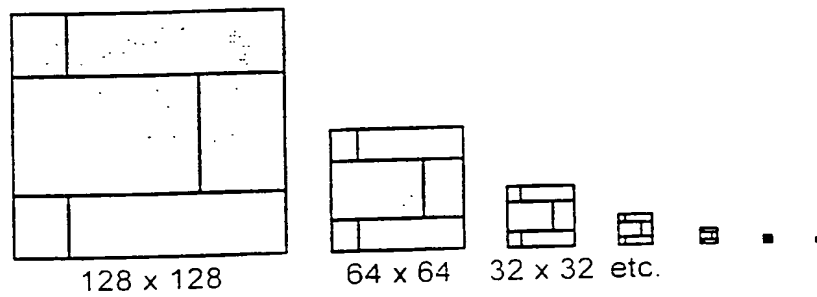


FIG. 3

2 / 9

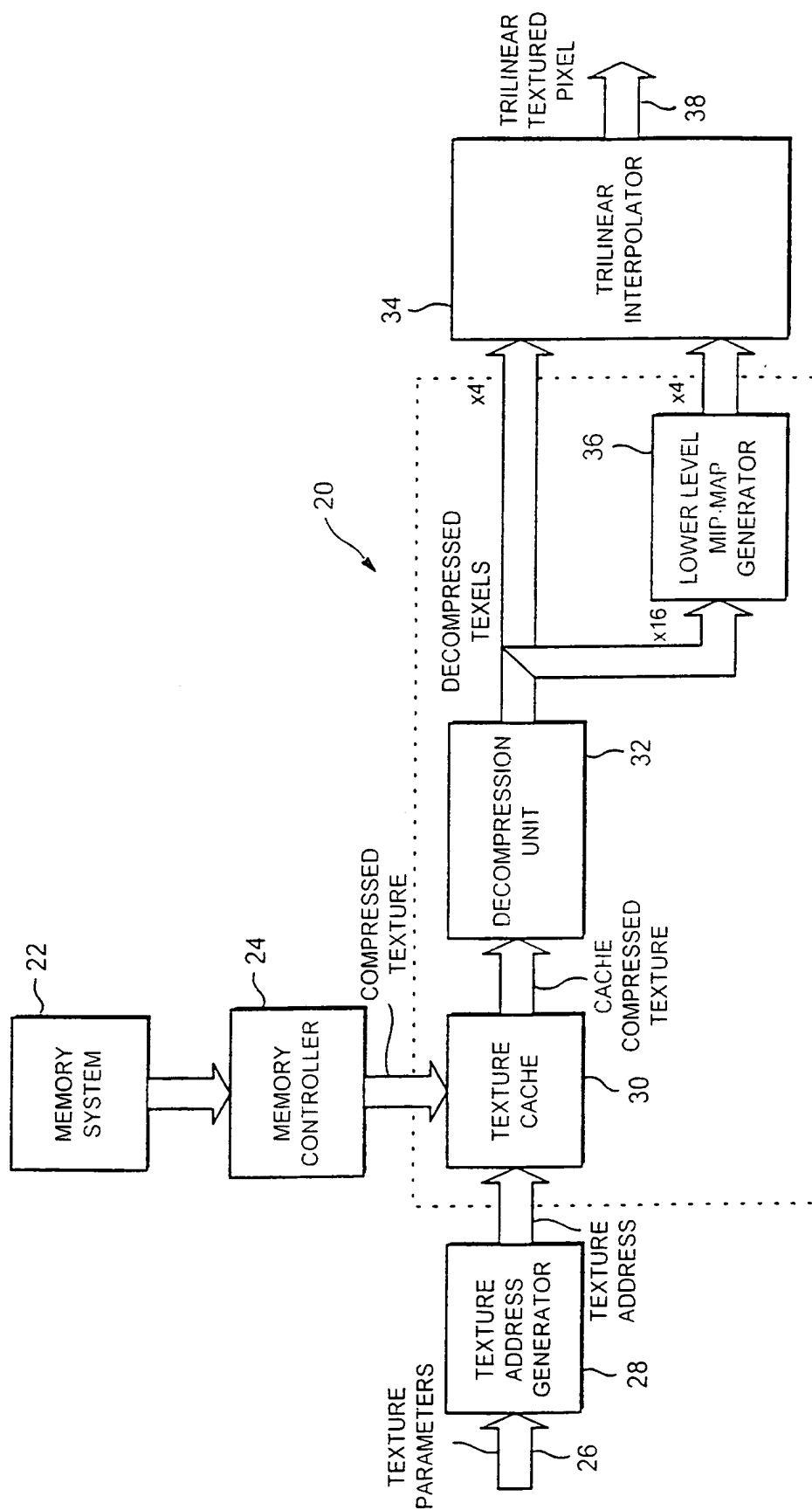


FIG. 4

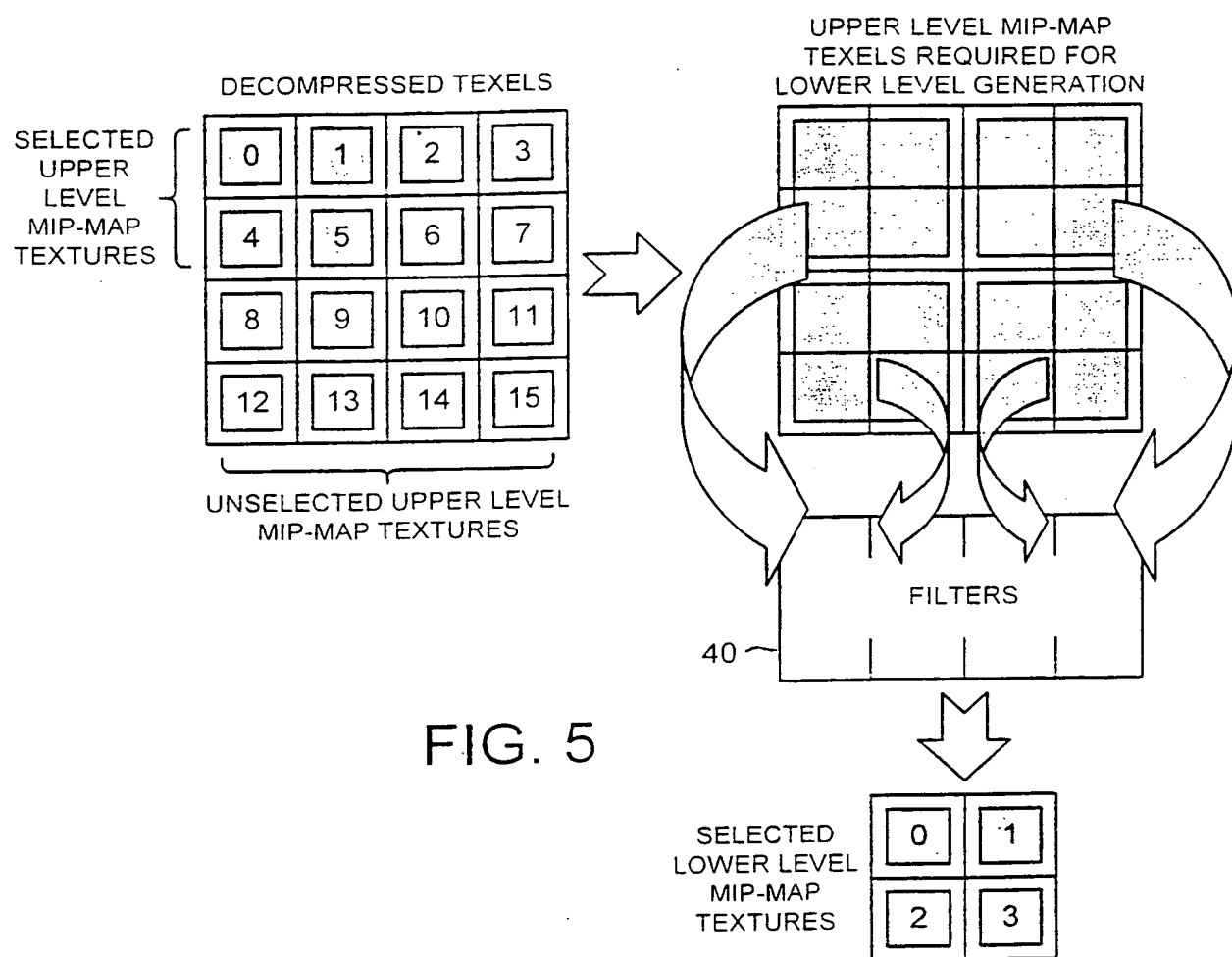


FIG. 5

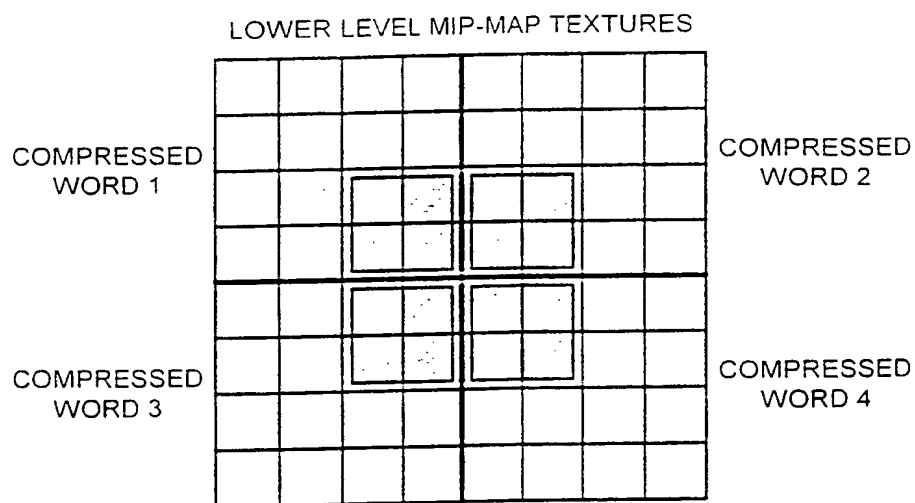
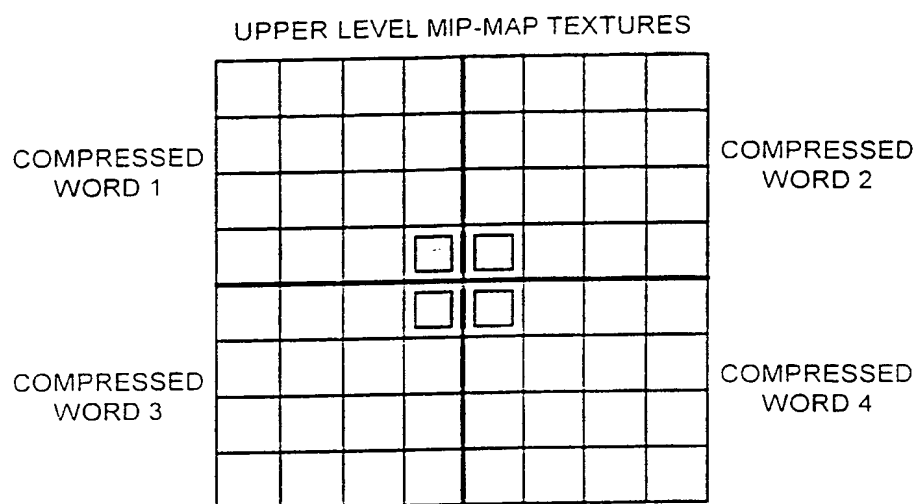
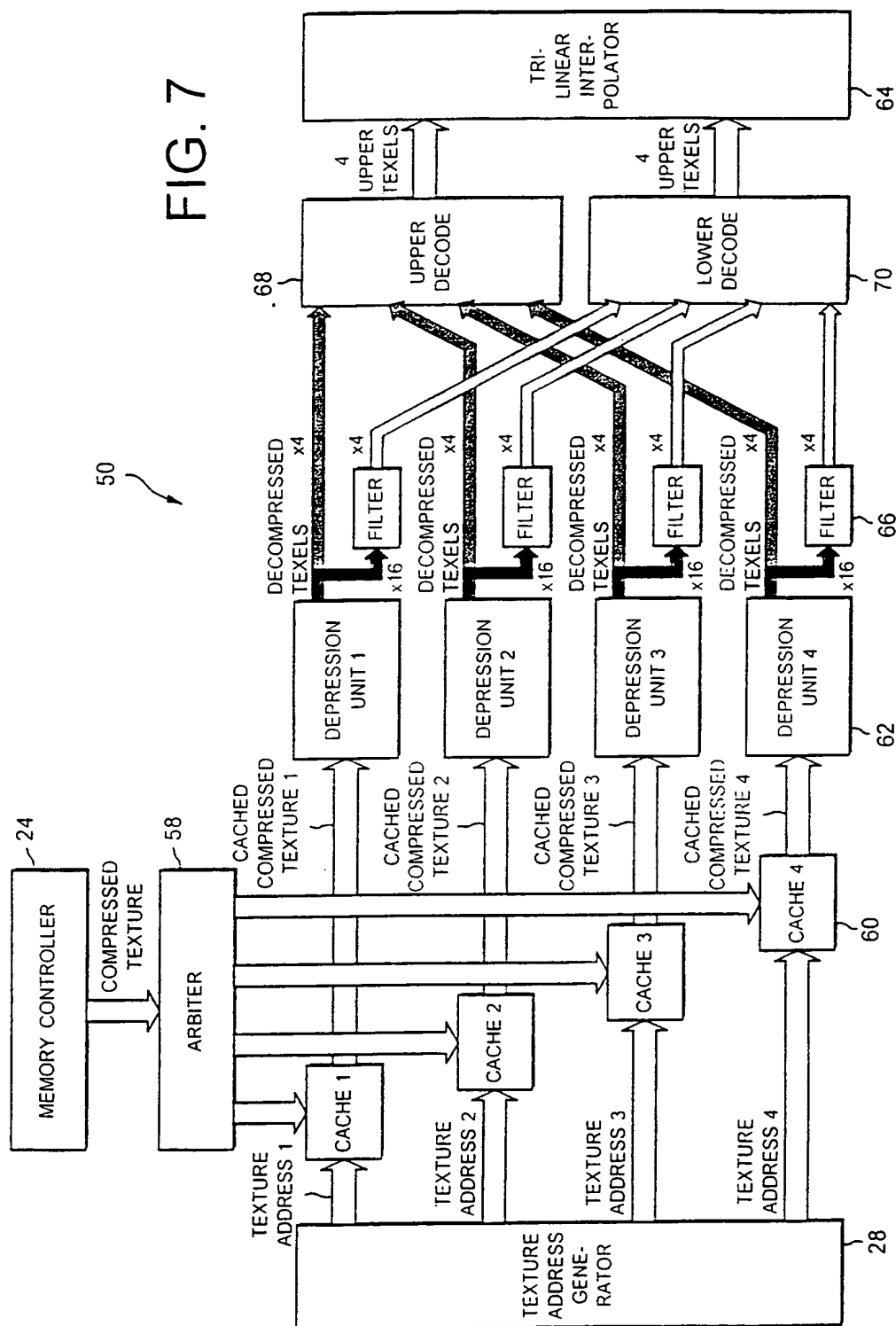


FIG. 6

FIG. 7



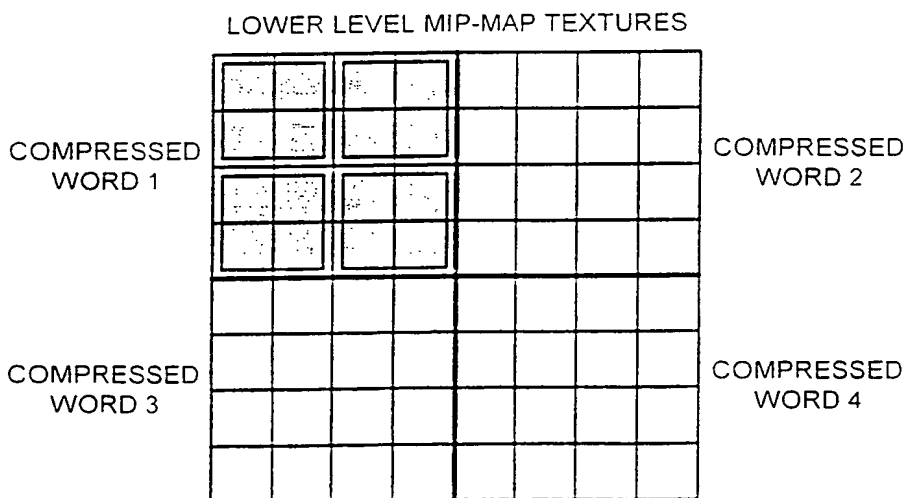
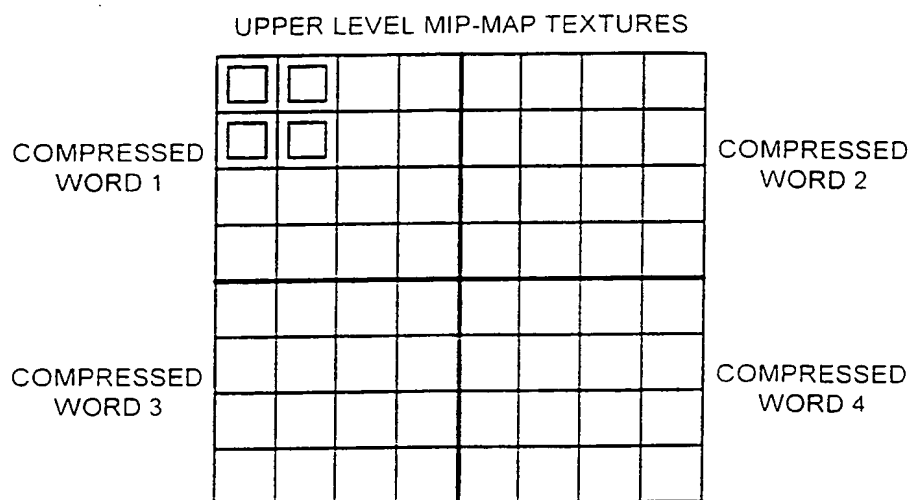
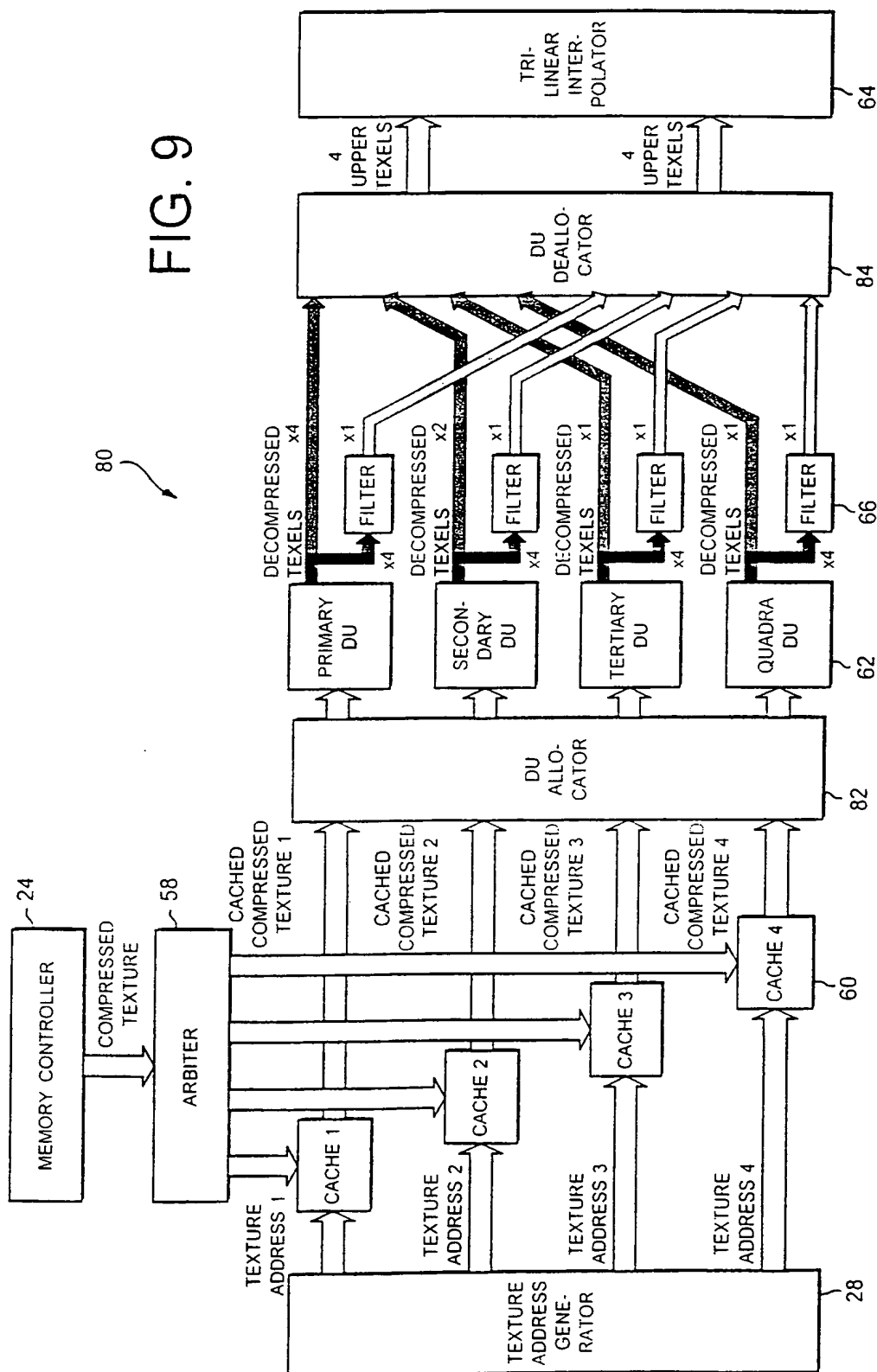


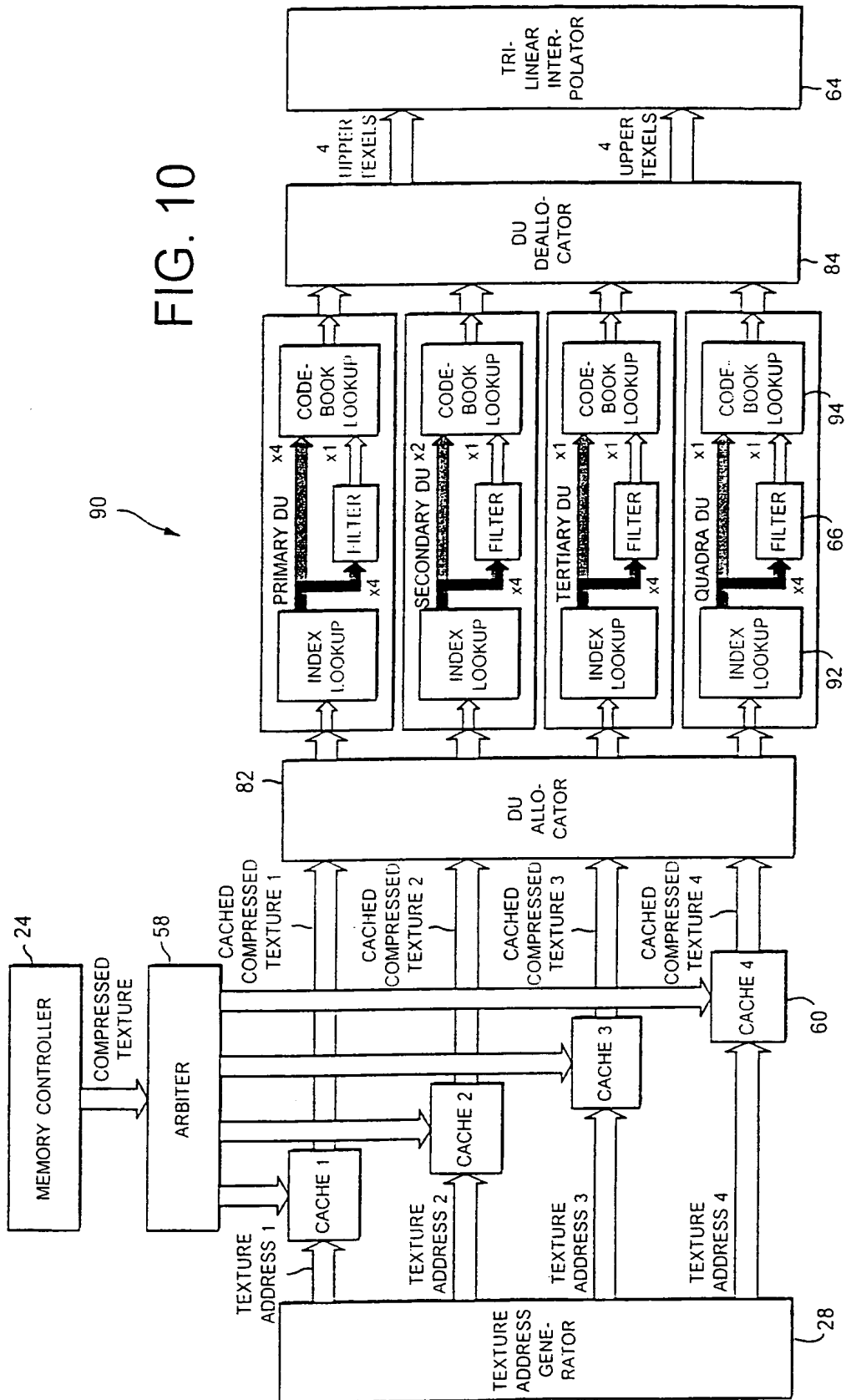
FIG. 8

FIG. 9



8 / 9

FIG. 10



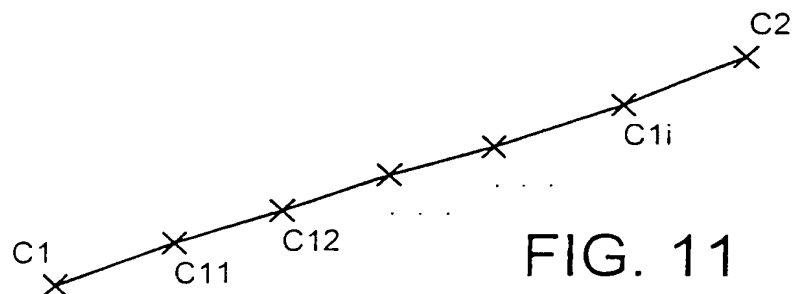


FIG. 11

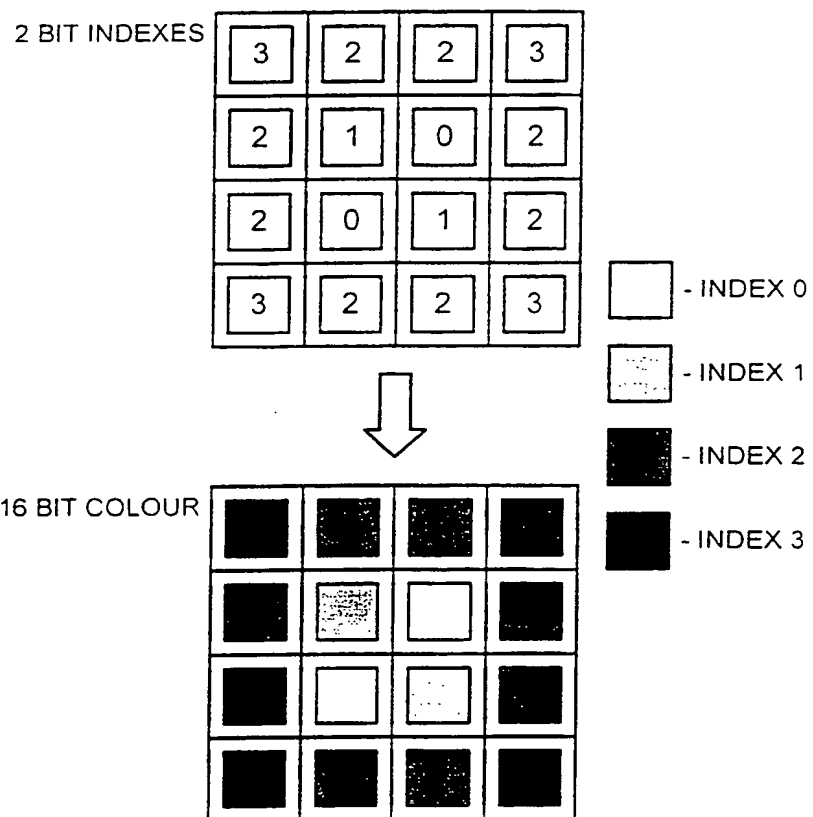


FIG. 12

INTERNATIONAL SEARCH REPORT

International Application No

PCT/GB 99/03711

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06T15/20

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 810 553 A (SEIKO EPSON CORP) 3 December 1997 (1997-12-03) ----	
A	WO 97 41534 A (3DFX INTERACTIVE INC) 6 November 1997 (1997-11-06) ----	
A	US 5 651 104 A (COSMAN MICHAEL A) 22 July 1997 (1997-07-22) -----	

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

3 February 2000

Date of mailing of the international search report

10/02/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Burgaud, C

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/GB 99/03711

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0810553 A	03-12-1997	US 5886705 A JP 10105723 A	23-03-1999 24-04-1998
WO 9741534 A	06-11-1997	US 5831624 A AU 2732797 A	03-11-1998 19-11-1997
US 5651104 A	22-07-1997	NONE	



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : G06T 15/20	A1	(11) International Publication Number: WO 00/28479 (43) International Publication Date: 18 May 2000 (18.05.00)
(21) International Application Number: PCT/GB99/03711 (22) International Filing Date: 8 November 1999 (08.11.99) (30) Priority Data: 9824407.2 6 November 1998 (06.11.98) GB (71) Applicant (for all designated States except US): IMAGINATION TECHNOLOGIES LIMITED [GB/GB]; Home Park Estate, Kings Langley, Hertfordshire WD4 8LZ (GB). (72) Inventor; and (75) Inventor/Applicant (for US only): PEARCE, Simon [GB/GB]; 66 Station Road, Kings Langley, Hertfordshire WD4 8LB (GB). (74) Agent: ROBSON, Aidan, John; Reddie & Grose, 16 Theobalds Road, London WC1X 8PL (GB).		(81) Designated States: JP, US, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report.</i>
(54) Title: TEXTURING SYSTEMS FOR USE IN THREE-DIMENSIONAL IMAGING SYSTEMS (57) Abstract A texturing system for use in a three-dimensional imaging system comprises a memory (22) for storing mip-map data for use in texturing an image. A controller (24) retrieves from the memory the mip-map data required and this data is stored in a cache (30). A lower-level mip-map generator (36) generates portions of the mip-map which is next below, in the hierarchical series, the mip-map of which portions are held in the cache. A trilinear interpolator (34) interpolates one output texel from input texels from the two mip-map levels. The texture data is represented by compressed codes. The lower-level mip-map generator (36) interpolates on the compressed code values.		

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

- 1 -

TEXTURING SYSTEMS FOR USE IN THREE-DIMENSIONAL IMAGING SYSTEMS

Field of the Invention

5 This invention relates to texturing systems for use in three-dimensional imaging systems, such as are used in computer displays. Such texturing systems are for example employed in the display of computer games.

Prior Art

10 The following prior documents and materials should be referred to for further information on the background to this technology. Certain of them are referred to where convenient by number in the subsequent description:

- [1] Catmull, E., "A Subdivision Algorithm for Computer Display of Curved Surfaces", Ph.D. Thesis, report
15 UTEC-Csc-74-133, Computer Sciences Department, University of Utah, Salt Lake City, UT, December 1974.
- [2] Blinn, J.F. and M.E. Newell, "Texture and Reflection in Computer Generated Images", CACM, 19(10), October 1976, 542-547.
- 20 [3] Wolberg, G., "Digital Image Warping", IEEE Computer Society Press, Los Alamitos, CA, 1990.
- [4] Williams, L., "Pyramidal Parametrics", SIGGRAPH 83, pages 1-11.
- [5] VideoLogic, "Apocalypse 3Dx data", available from
25 VideoLogic Limited, Kings Langley, England.
- [6] Microsoft Corporation, "Texture and Rendering Engine Compression (TREC)", Microsoft Technical Brief, internet address: www.microsoft.com/hwdev/devdes/whntrec.htm
- [7] Beers, A. C. Agrawala, M. and Chaddha, N., "Rendering
30 from Compressed Textures", 1996 Computer Graphics Annual Conference, page 373.

- 2 -

[8] Hakura, Z.S. and Gupta, A. "The design and analysis of a cache architecture for texture mapping", Computer Architecture News, Vol.25, no.2, pp.108-20, May 1997.

[9] United States Patent US-A-5,612,747 "Method and apparatus for vector quantization caching in a real time video coder".

[10] United Kingdom Patent Application GB-A-2,297,886 "Texturing and shading of 3-D Images ", Applicants: VideoLogic Limited.

[11] Gray, R. M., "Vector Quantization", IEEE Transactions on Communications, January 1980, pp.4-20.

[12] Koegel Buford, J., "Multimedia Systems", Addison-Wesley publishing company, 1994.

[13] Foley, F. and van Dam, A., "Computer Graphics Principles and Practice", Addison-Wesley publishing company, 1990.

[14] Microsoft, "DirectX 6.0 SDK Documentation", Microsoft Corporation, 1 Microsoft Way, Redmond, USA. Internet address: www.microsoft.com/directx.

Background of the Invention

Computer based images are commonly formed of an array of picture elements or pixels. Each surface to be displayed may be represented by the pixels within a polygon, commonly a triangle. The surface is given color, texture and/or shading by an operation known as "texture mapping". Textures are stored as arrays of pixels, conveniently termed texels (texture pixels). Thus "texture mapping" involves the mapping of a 2D (two-dimensional) array of texels onto another 2D array of pixels representing a surface in a 3D (three-dimensional) scene. This technique was developed by Catmull [ref. 1] and refined by Blinn and Newell [ref. 2]. Perspective texture mapping involves rotating and translating the texture map so that its perspective appears correct on the

- 3 -

final image. Texture mapping improves the realism of the scene by giving the surface of an object a realistic finish. An example of this is mapping a marble texture to the surface of a statue, giving the statue the appearance that it is made of marble. For a large scene many different texture bitmaps are required to represent all the different textures which might be present in the scene.

As just noted, a 3D scene is usually represented by a number of polygons or triangles. In order to fill a polygon with a texture, each pixel on its surface is used to calculate the co-ordinate of a texel in the texture map. The nearest texel to the one calculated in the texture map may be used to shade the finally displayed pixel. This is called point sampling. Alternatively, bilinear filtering or bilinear interpolation may be used to improve the quality of the textured image. In bilinear filtering the point in the texture map from which the 2D pixel is to be mapped onto the 3D surface is calculated to sub-pixel accuracy. Bilinear filtering or interpolation is then used to blend the four closest pixels to this calculated position in the texture map in order to attain a more accurate representation of the pixel color. This is illustrated in the accompanying Figure 1, where the texels A, B, C, and D are blended to provide a texel value for a pixel at point X on the two-dimensional image plane. This operation of bilinear, i.e. two-axis, filtering (or interpolation) is further described in ref. 3.

Trilinear (three-axis) filtering is the same process over the four closest pixels on two different mip-map levels [ref. 4]. This is illustrated in Figure 2 of the present application. Mip-maps are copies of the original texture map which have been pre-processed by being filtered so as to be successively reduced to half the resolution. MIP here stands for MULTUM IN PARVO (much in

- 4 -

a small place). This is repeated until the resulting image is 1 pixel in size (this assumes that the texture is square and of a size which is a power of 2), so that there are a hierarchical series of the mip-maps. Figure 3 shows
5 an example of a brick texture at 128 x 128 resolution with the associated lower mip-map levels. A mip-map can be thought of as a pyramid.

Texture filtering has the effect of reducing the occurrence of aliasing when sampling textures. For more
10 information on aliasing see ref. 3.

Three-dimensional image generation is computationally intensive. Animated 3D images for games and Computer Aided Design (CAD) applications are becoming increasingly expensive in terms of processing power, as scenes become
15 more photo-real and images are required to respond in real-time. A large number of floating point calculations are required to determine the geometry of the polygon structure in the scene and a large number of arithmetic operations are required to fill and shade the polygons.
20 Dedicated hardware is available [ref. 5] that can perform these operations many times more efficiently than software. Accesses to stored databases are also a limiting factor to performance. Local memory in dedicated hardware can reduce the effect of any memory access
25 bottlenecks. Texture mapping is particularly memory intensive especially when performing a filtering (that is, interpolation) operation where many texture pixels are read for every pixel that is mapped onto the display.

The size of a 2D texture map data is therefore
30 reduced by texture compression so that it can be located into a smaller memory space. A small memory requirement leads to lower system costs. The original texture map can then be retrieved from the compressed data by decompression. As 3D scenes become more realistic,
35 texture maps become larger and more numerous, making the

- 5 -

use of texture compression more important. Several schemes have already been developed including Texture and Rendering Engine Compression (TREC) from Microsoft [ref. 6]. Beers [ref. 7] first discussed the technique of rendering images from compressed textures.

It is convenient at this point to consider, and define, the various types of memory that are available to the system designer. The term "local memory" refers to solid state semiconductor memory located close to the memory control semiconductor device or circuit. The term "internal memory" refers to memory located within the particular semiconductor device being referred to.

"External memory" is any memory outside the semiconductor device. Local memory can be DRAM based. DRAM is an acronym for Dynamic Random Access Memory, which is a solid-state semiconductor. Synchronous DRAM (SDRAM) enables data accesses to be co-ordinated by a clock signal. SDRAM has a higher access bandwidth capability than DRAM due to its pipelined architecture but is more expensive. Local memory and internal memory can be DRAM or SDRAM based. External memory can be sold-state or a mass storage array such as a hard disk. Semiconductor memory is very expensive and makes up a large percentage of the overall cost of a computer system.

DRAM is addressed over a multiplexed address bus, that is, the address needed to access an individual data item is transmitted to the memory device in two parts. The core memory array in the DRAM device is a rectangular matrix where a single data item is addressed when a row control line and a column control line are activated at the same time. This requires a separate row and column address. If the row address does not change between sequential accesses, then only the column address needs to be transmitted. A row of data in the DRAM array is known as a page. When the row address remains unchanged between

- 6 -

accesses, the accesses are said to be "in page". "In page" accesses are much quicker than those that span two or more pages, and memory system designers endeavour to keep bursts of accesses in page. Some memory devices, such as SDRAM, make use of multiple memory banks to improve performance. Each memory bank can have its own page open, permitting data accesses to separate areas of memory without breaking page.

One technique used to improve memory performance is "Memory Caching" in which the result of all external memory accesses is stored in a separate internal memory. This internal memory can be accessed much faster than external memory. If the result of a particular memory access already resides in the cache, then the cache is read instead of the external memory. This has the effect of reducing traffic to the external memory, and therefore reducing the "bandwidth" requirements of that memory. The bandwidth requirement of a memory system is often directly related to the cost of that system. In a fixed bandwidth system an increased bandwidth requirement can lead to a reduction of overall system performance.

Texturing is the most performance-intensive process of 3D imaging as it requires all textures to be retrieved from memory. Techniques such as trilinear and bilinear filtering (interpolation) require up to eight texture pixels or texels to be retrieved from memory for every pixel projected onto the display, as described above and illustrated in Figures 1 and 2. Texturing therefore requires a very high bandwidth path into memory. Texture caching can be employed to reduce the texturing bandwidth requirement and increase system performance. The optimum performance objective is to be able to read all necessary texels in one processing pipeline clock cycle. Some work has already been done on studying the effects of using a cache to improve the performance of texture accesses

- 7 -

[ref. 8]. Hakura demonstrates how caches can be highly effective for texture mapping graphics, and concludes that the effective memory to bandwidth ratio can be reduced by a factor of three to fifteen using certain caching strategies.

As previously indicated, texture mapping is used to improve 3D image quality by mapping high detail onto a surface of a 3D object. This should be done without complicating the object's geometry. However, texture mapping produces a wide variety of visual artefacts, including aliasing [ref. 13]. Bilinear filtering [ref. 3] is used to improve the quality of the resulting image but there remain many artefacts that bilinear filtering cannot solve, including depth aliasing. Depth aliasing is the result of the texture getting more compressed as an object moves further from the viewpoint. This form of aliasing can be resolved by use of mip-maps [ref. 4], but there is still a problem called mip-banding. Mip-banding occurs during the transition period between mip-maps when the texture changes from one level of detail to another. This may appear for example on a road, seen in the foreground, which disappears into the distance. Successive mip-maps are used along the road and the transition from one mip-map to the next can be visible. This problem can be solved with the application of trilinear filtering [ref. 4], which interpolates the level of detail between mip-maps, as described above.

The best form of trilinear filtering is that which is performed on a per-pixel basis. This requires eight texture pixels (texels) to produce the final on-screen pixel. As these texels can be located anywhere in memory, eight separate memory reads are often required. Trilinear filtering is performed between two mip-levels, and so four memory reads occur from one mip-map location and four from another. Textures are usually stored in local memory,

- 8 -

although system memory texturing is becoming more popular. These memories have a finite bandwidth and are very often required to serve as a resource to memory for many different applications. Set-up parameters, depth
 5 information, and display information are usually stored in local memory, and system applications are usually run from system memory. Eight individual memory reads per pixel is usually beyond the capabilities of many memory systems. Added to page change between mip-maps, this often achieves
 10 less than adequate 3D performance.

The memory bandwidths required for a trilinear texture access system is dependent on the number of memory accesses needed for each texture filtering operation and the pixel throughput performance demanded by the
 15 application. Equation 1 shows how the texture bandwidth can be determined. The equation also shows how the bandwidth of page breaks must also be taken into account.

$$Bandwidth_{texture} = ((Accesses_{pixel} \times Width_{memory}) + (Accesses_{page_break} \times Width_{memory})) \times Throughput_{pixel}$$

... Equation (1)

Where:

20 $Bandwidth_{texture}$ is the texture bandwidth demanded from the memory measured in bytes/s. This is not the memory bandwidth that can be supplied by the memory.

$Accesses_{pixel}$ is the average number of memory accesses per pixel. Not all the required texels can be read in
 25 one access, even with the right data width.

$Accesses_{page_break}$ is the average number of memory access slots lost to page breaks per pixel. A single page break using SDRAM requires at least 8 accesses slots.

$Width_{memory}$ is the width of the memory data bus, measured in
 30 bytes. This has to be at least 8 bytes (64 bits) to

- 9 -

ensure that four texels can be read in one clock cycle.

$Throughput_{pixel}$ is the pixel throughput demanded by the application, measured in pixels/s. For most modern applications this is around 100 Mpixels/s.

The average accesses per pixel is the number of separate memory accesses required to retrieve all data necessary for the filtering operation. Using a 64-bit memory bus, maximum throughput is achieved if four 16-bit texels are required and they reside in the same data word. Unfortunately this is not always the case and very often the texture data resides in two or four separate words. Equation 2 shows how $Accesses_{pixel}$ can be found, taking into account the varying number of accesses for a single texture operation.

$$Accesses_{texture} = \frac{((Percentage_{single} \times 1) + (Percentage_{double} \times 2) + (Percentage_{quadruple} \times 4) \times Mipmaps)}{Factor_{compression}}$$

... Equation (2)

Where:

$Percentage_{single}$ is the percentage of single mip-map accesses that can be retrieved in one memory access.

$Percentage_{double}$ is the percentage of single mip-map accesses that can be retrieved in two memory accesses.

$Percentage_{quadruple}$ is the percentage of single mip-map accesses that can be retrieved in four memory accesses.

$Mipmaps$ is the number of mip-maps involved in the filter operation.

$Factor_{compression}$ is the compression factor if the texture is compressed.

- 10 -

The average number of accesses is calculated by taking into account the likelihood that all data will be retrieved in single, double or quadruple accesses. The equation also takes account of the number of mip-maps
5 required in the filtering operation, trilinear filtering uses two but bilinear requires only one. The equation also shows how the average number of accesses is reduced linearly with compression. Obviously the less data there is to fetch the less memory accesses that are required and
10 the more likely all the data will reside in the same data word.

Equation 1 thus shows that, even with the use of texture caches, the physical memory bandwidth requirement still remains beyond the scope of any viable memory
15 system. For this reason texture compression is employed, not only to reduce the physical size of the stored texture, with its associated reduction in memory cost, but also to reduce the volume of texture data that is transferred from that memory.

Equation 2 shows how the compression factor affects the number of memory accesses. High performance, dedicated hardware can be used to decompress the textures in real-time after they have been read from memory. Many texture compression techniques have been tried, including:
25 Vector Quantisation (VQ)[ref. 11]; Color lookup tables (CLUT) or Palletisation [ref. 12]; Discrete Cosine Transformation (DCT)[ref. 12]; and several proprietary techniques [refs. 6 and 14]. But each has its associated problems: VQ and Palletised require two memory reads or
30 large internal data caches and quality can be limited; DCT requires a large amount of decompression logic, and with a limited silicon budget this can be unfeasible; and many proprietary techniques provide limited compression ratios and quality. As Equation 1 demonstrates, these techniques

- 11 -

only go part way towards resolving the bandwidth requirements of trilinear filtering.

Memory access streams that continuously swap between different memory banks can have a large effect on performance. Equation 1 shows how dominant page breaks are to the performance of texture filtering as a whole. For trilinear filtering, page breaks can be particularly problematic, where a number of mip-maps can often span more than one memory page.

3D imaging techniques often demand such a high level of performance that only dedicated hardware solutions can be used. This often requires the development of a special silicon chip. As well as performing texture mapping, the chip will often be called upon to perform all the geometry processing, polygon set up, image projection, illumination calculations, fogging calculations, hidden surface removal, and display control. Therefore it is critical that each stage in the generation of a 3D image is made as small as possible to enable all processes to fit on the same silicon die. As well as requiring a large memory bandwidth, a trilinear filtering operation can only be implemented in a large amount of logic and therefore silicon area. It is important that an optimum solution be found that limits the required logic, and therefore cost, to a minimum.

It is seen from the foregoing that the requirements of memory, speed and ease of construction of the chip are very substantial and are taxed to the full in 3D imaging, particularly when texture mapping. Even using all available techniques for meeting the requirements, the constraints are still very difficult to meet if high quality real-time imaging is to be achieved.

- 12 -

Summary of the Invention

The invention in its various aspects is defined in the independent claims below, to which reference should now be made. Advantageous features are set forth in the
5 appendant claims.

Preferred embodiments of the invention are described below with reference to the drawings. In these embodiments the efficiency of trilinear texture filtering is improved by the generation of the lower-level texture
10 mip-map on the fly from the upper-level mip-map, and with the use of texture caching and texture decompression as a means of meeting the high memory bandwidth requirements. Removing the need to read the lower-level mip-map in a separate memory access, removes page break problems and
15 enhances performance.

More particularly, the preferred texturing systems comprise a memory for storing mip-map data for use in texturing an image, the mip-map data comprising a hierarchical series of mip-maps of different levels of
20 decreasing resolution. Data is received at an input indicating the type of mip-map data required and the level of the mip-map or mip-maps from which the data is to be taken. A controller retrieves from the memory the mip-map data required in accordance with the input data, and this
25 data is stored in a cache. A lower-level mip-map generator generates portions of the mip-map which is next below, in the hierarchical series, the mip-map of which portions are held in the cache. A trilinear interpolator receives from the cache mip-map data from one level of
30 mip-map and from the lower-level mip-map generator data from the next lower level of mip-map, and interpolates one output texel from input texels from the two mip-map levels.

Some data compression schemes lend themselves well to
35 the generation of lower mip-maps on the fly by providing

- 13 -

the required quantity of decompressed data and by closely supporting the filtering algorithm used to generate the lower-level mip-map. These techniques may be used to enhance the performance and quality of a 3D rendered image while minimising the hardware overhead.

In one preferred embodiment, the texture data is represented by arbitrary compressed codes, in which selected compressed code values define principal colors and other compressed code values define intermediate colors which can be formed by selected weighted averages of principal colors, the corresponding code values also being weighted averages of the code values of the selected principal colors. The lower-level mip-map generator interpolates an output texel from a plurality of input texels by operating on the compressed code values.

Brief Description of the Drawings

The invention will now be described in more detail, by way of example, with reference to the drawings, in which:

Figure 1 is a diagram illustrating bilinear filtering of the texture map in an imaging system;

Figure 2 is a similar diagram illustrating trilinear filtering of the texture map in an imaging system;

Figure 3 illustrates a series of mip-maps;

Figure 4 is a block schematic diagram of part of an imaging system in accordance with the present invention;

Figure 5 is a diagram illustrating the operation of the embodiment of Figure 4;

Figure 6 illustrates the storage of texture words in a worst-case scenario;

Figure 7 is a block schematic diagram similar to Figure 4 of a second embodiment of the invention using four caches and four decompression units in parallel;

- 14 -

Figure 8 illustrates in similar manner to Figure 6 the storage of texture words in a best-case scenario;

Figure 9 is a block diagram of a modification of the embodiment of Figure 7, in which the decompression units
5 are dynamically re-allocated;

Figure 10 is a block diagram of another modification of the circuit of Figure 7; and

Figures 11 and 12 illustrate how interpolation can be directly applied to compressed texture codes.

10 Detailed Description of the Preferred Embodiments

Figure 4 shows a specific embodiment of a texturing system in accordance with the invention. The texturing system 20 comprises a memory system 22 forming part of the memory used by the imaging system as a whole, and which is
15 controlled in known manner by a memory controller 24. The memory 22 holds *inter alia* a compressed texture map, and the output of the memory through the memory controller 24 consists of compressed texture from the requested part of the texture map. Parameters defining the texture required
20 for any surface are received at an input 26 and applied to a texture address generator 28 to address the part of the memory 22 where the desired texture is located. The compressed texture is retrieved from memory by the memory controller 24 and held in a texture cache 30. The
25 retrieved texture will correspond to a particular texture type representing the type of surface to be displayed. For example this may be part of a brick wall or the surface of a road. This surface texture will be held in mip-map form, as described above, actually in compressed
30 mip-map form, and the mip-map which is used will be the one which is the next in front of the surface, account being taken of the distance of the surface from the observer.

- 15 -

The output from the texture cache 30 is applied to a decompression unit 32, which decompresses or decodes the texture values to provide decompressed texels. The decompressed texels are then applied both to a trilinear interpolator 34 and to a generator 36 for generating the relevant part of the next lower mip-map to that held in the texture cache 30. The output of the generator 36 is also applied to the trilinear interpolator 34. Thus the interpolator receives the texels from the mip-map in front of and that behind the surface point under consideration, so that a trilinear interpolation, as illustrated in Figure 2, can be executed. The trilinear interpolator provides a trilinear-textured pixel at an output 38.

In operation, therefore, the texture address generator 28 feeds the addresses of the eight trilinear texels to the texture cache 30, which then fetches the associated compressed texture data from memory 22. The decompression unit (DU) 32 then decompresses 16 upper mip-map texels from the compressed data. Four are selected as the requested upper-level texels, and all 16 are used by the lower mip-map generator 36 to generate the lower-level mip-map texture. This is because four upper-level texels will be used to generate each lower-level texel, as described below. These upper-level and lower-level mip-maps are then used by the trilinear interpolator 34 to generate the single trilinear pixel. Although as many as 16 texels are required for this technique to work, which, uncompressed, would represent a large memory bandwidth, when compressed this represents a considerable memory bandwidth improvement with minimum page breaks.

The lower-level mip-map is generated by four parallel filters or interpolators 40 that are able to take the 16 upper-level texels and generate the four lower level in one clock cycle. Figure 5 illustrates this process. Out

- 16 -

of the 16 original texels, four are decoded to produce the four upper-level texels which are directly used for trilinear filtering. These 16 are also divided into four quadrants, each containing four pixels, which are then
5 filtered by four digital low-pass filters to produce the four lower-level mip-map texels required for trilinear filtering. This filtering algorithm is shown in Equation 3:

10 Lower-level color value0 = (upper0 + upper1 + upper4 + upper5) / 4
Lower-level color value1 = (upper2 + upper3 + upper6 + upper7) / 4
Lower-level color value2 = (upper8 + upper9 + upper12 + upper13) / 4
Lower-level color value3 = (upper10 + upper11 + upper14 + upper15) / 4
... Equation (3)

By applying actual values to Equation 1 and Equation
15 2, the cost of trilinear filtering can be determined. It has been determined that in a conventional system, four accesses of 64-bit data are required for each trilinear pixel when uncompressed, with an average of two page breaks per pixel. This gives a total bandwidth
20 requirement of 16 Gbytes/second for a system that must produce 100 Mpixels per second. Such performance is unacceptable, but even when the texture data is cached and compressed the bandwidth demanded is not much better (13.2 Gbytes/s). However, when the double page break is
25 removed by generating the lower mip-map "on the fly" in the generator 36 of Figure 4, in accordance with this invention, the bandwidth is reduced to an acceptable level. Cached and compressed texture bandwidth requirements for this system are of the order of only
30 200 Mbytes/s. A 64-bit memory system running at 100 MHZ has a peak bandwidth capability of 800 Mbytes/s, so the required 200 Mbytes/s is well within its capacity. Indeed, the lower-level mip-map construction system

- 17 -

embodying the invention can leave spare bandwidth which will be available for other uses.

In summary, therefore, it will be seen that the texturing system of Figure 4, which is designed for use as part of a three-dimensional imaging system, includes the memory 22 for storing mip-map data for use in texturing an image, the mip-map data comprising a hierarchical series of mip-maps of different levels of decreasing resolution. The input 26 receives input data indicating the type of mip-map data required, and the level of the mip-map or mip-maps from which the data is to be taken. The controller 24 is coupled to the input and to the memory for retrieving from memory the mip-map data required in accordance with the input data. The cache 30 is coupled to the controller for storing portions of mip-map data retrieved from memory which relate to a selected mip-map level. The trilinear interpolator 34 is coupled to the cache to receive mip-map data from one level of mip-map, namely the upper level of the two levels from which data is needed, and to interpolate an output texel from input texels from two mip-map levels. In accordance with the invention, the texturing system also includes the lower-level mip-map generator 36 which is coupled between the cache and the trilinear interpolator, and generates in real-time portions of the mip-map next below (in the hierarchical series) the mip-map of which portions are held in the cache.

The embodiment of the invention illustrated in Figure 4 shows how one cache 30 and one decompression unit (DU) 32 are required to return all data necessary to construct a complete trilinear interpolated pixel. A problem exists, however, when the filter algorithm needs data from two or four additional data words. Figure 6 shows the worse case situation. Consider the texture bitmap divided into tiles, with each tile representing the

- 18 -

texture pixels provided by one compressed data word. In this example the data word contains 16 texture pixels. The darker outlined boxes in Figure 6 represent the tiles. In this example, the filter algorithm requires texture
5 pixels from four separate compressed data words. For a single cache/decompression system, this worst case situation requires four cache reads and therefore a minimum of four clock cycles. This problem also exists,
10 to a lesser extent, when two separate data words are required.

A four-cache and four-DU construction can provide all requisite data in one clock period, no matter where the compressed data resides. Figure 7 shows a block diagram of a second embodiment of the invention which comprises
15 such a 'quad cache' arrangement. The texturing system of Figure 7 is similar to the system of Figure 4 and only the differences will be described in detail. In the second embodiment of Figure 7, the memory system 22 and input 26 are the same as in Figure 4. In this case the texture
20 address generator 28 has four outputs to cope with the situation where the outputs of a maximum of four tiles are required. The four outputs provide respectively texture addresses 1 to 4, and these are applied to texture caches 1 to 4, referenced 60. These communicate to and from the
25 memory controller 24 through an arbiter 58, so as to receive the texels from the four tiles respectively. These are held in the four caches, which thus output four cached compressed textures, namely cached compressed texture 1 to 4 respectively.

30 The outputs of the four caches 60 are respectively applied to four decompression units 62, each similar to the decompression unit of Figure 4. Each decompression unit 62 applies its decompressed data to upper and lower decode units 68, 70, and also to a low-level mip-map filter

- 19 -

66. The outputs of the upper and lower decode units 68,70 are respectively applied to a trilinear interpolator 64.

In operation, each cache 60 is responsible for a different segment of the memory, based on the size of a cache word and the lower two bits of the texture address representing each cache word. In this way it can be guaranteed that a single cache will not be called upon to provide more than one data word for a complete trilinear pixel.

Without being able to generate the lower-level mip-map from the upper one, eight caches and DUs would be required, four for the upper map and four for the lower map. Thus, the generation of lower-level mip-maps as required 'on the fly' saves logic (silicon area) as well as improving performance. When coupled with compression, the amount of data required is also greatly reduced.

It will be appreciated that the lower-level mip-maps are still stored in the memory. They will be required for pixels which are yet further away from the viewer.

However, it proves to be more efficient in the trilinear interpolation to ignore their presence and to regenerate those texels which are needed for the pixels being processed using the generators 62.

One of the major advantages of generating lower-level mip-maps from compressed upper-level mip-maps is the amount of logic optimisation that can take place. Less logic leads to smaller hardware and lower cost. One such area of optimisation is in the caches and decompression units.

The caches 60 for the quad cache system are effectively memory mapped, i.e. each cache holds data for a separate tile of texture memory, as noted above. However, the DUs 62 do not necessarily have to be memory mapped as well. Memory mapping of DUs would require that each is associated with a particular cache. This would

- 20 -

also necessitate each DU being able to produce at least four upper-level texels as well as four lower-level texels in one cycle to cater for the best case situation. The best case situation is illustrated in Figure 8 where all
5 the tiles come from one data word.

By allowing each DU to be dynamically reallocated based on the resources required, then several hardware optimisations can take place to dramatically reduce the size of the hardware required. We have determined that
10 not more than one unit need provide four parallel upper-level texels, not more than two units need provide two parallel upper-level texels, and none need provide more than one lower-level texel.

Figure 9 shows how these optimisations fit into a modified quad cache system. The modified arrangement of
15 Figure 9 is similar to the arrangement of Figure 7 and only the differences need be described. These are the inclusion of a decompression unit (DU) allocator 82, which is connected between the outputs of the four caches 60 and the inputs of the four decompression units 62, and a
20 corresponding deallocator 84, which replaces the upper and lower decode units 68,70 and is connected between the outputs of both the four DUs 62 and the filters 66, and the inputs to the trilinear interpolator 64. The DUs 62
25 are no longer identical; the primary DU can provide four parallel upper-level texels, the secondary DU can provide two parallel upper-level texels, and the tertiary and quadra DU can each provide only one upper-level texel. The DU allocation unit 82 determines which of the DUs 62 are
30 required, based on the configuration of the texture addresses. Logic reduction takes place both on the registered outputs of each DU and the multiplexing of the texture output from each DU. The deallocation unit 84 now
35 takes over from the upper and lower decode units 68,70, this time returning the dynamically allocated resource

- 21 -

data to the data paths that originally requested the texels. The arithmetic used to filter the four lower-level mip-map texels is also simplified.

5 The outputs are shown as being deallocated by the deallocator 84, but this may not be necessary depending on the manner in which the output of the trilinear filter 64 is handled. This is an implementational detail which will be clear to the person skilled in the art.

10 Another area of logic optimisation is in the filters 40 (Fig. 5) that generate the lower-level mip-map. If the compression scheme is of a type which uses an index to look up a code book, such as in VQ or Palletised (see above), then in this case the texture components can be arranged in such a way as to enable ease of lower mip-map
15 generation. That is, if the code book entries are generated by evenly sampling the uncompressed texture over its color range, the indexes themselves can be used as the input to the mip-map filters to generate the new lower-level mip-map. The resulting index will then point
20 to the closest code book equivalent of the filtered texture. The bit width of the filtering arithmetic on the indexes (compressed texture) is much smaller than had the filter been applied to the uncompressed texture.

25 Such an arrangement is illustrated in Figure 10 of the drawings. This is based on Figure 7, though it could of course also include the modification of Figure 9.

Figure 10 shows how the filter 66 now appears inside the decompression unit 62. Each decompression unit 62 has an index lookup section 92 connected to the output of the
30 allocator 82, to provide indices which are filtered by the filter 66 before being applied to a codebook lookup section 94 where the filtered indices are used to find the corresponding codebook portion.

35 Figure 11 diagrammatically illustrates the relationship between the compressed code values and the

- 22 -

colors they represent. The diagram is assumed to be a part of the color triangle representing all possible colors on a two-dimensional plane. Two colors C1 and C2 are denoted as principal colors. They define the ends of a line on which other intermediate colors C11 to C1i lie. These other colors can therefore be formed by taking an appropriate weighted average of the colors C1 and C2. The code values for the colors C11 to C1i are likewise defined by averaging the code values of the colors C1 and C2.

The normal way of producing the intermediate colors is to start with the compressed code values of the colors C1 and C2 and to determine the actual color values of the colors C1 and C2 to the accuracy of the system, which may require 16 or even 24 bits per color. A weighted average is then performed on the 16 or 24 colors values thus obtained. However, we have appreciated that when the relationship between the colors and their code values is defined in the way just described, it is possible to average the code values first and then subsequently to convert the averaged code value thus obtained to the nearest available color. This means that the averaging operation no longer has to be done with 16 or more bits of data, but can now be done with just the two or so bits which are required to define the different compressed code values in use.

This lower mip-map optimisation technique can also be applied to Direct X compressed textures [ref. 14]. Here the two primary colors and the interpolated intermediate values pointed to by indexes are spaced evenly apart. Figure 12 shows how 2-bit indexes are mapped to 16-bit texture colors. It is seen that the 2-bit indexes themselves can be filtered instead of the full 16-bit value. This represents much less logic and therefore less overall cost.

- 23 -

In summary, the illustrated method of generating texture data for use in texturing an image, is seen to comprise first representing texture data by arbitrary compressed codes, in which selected compressed code values define principal colors and other compressed code values define colors which can be formed by selected weighted averages of principal colors, the corresponding code values also being weighted averages of the code values of the selected principal colors. Then an output texel is interpolated from a plurality of input texels, with the interpolating step being effected using compressed code values. The code values are subsequently decompressed to give the actual color values.

This method finds application for other purposes than just in the lower-level mip-map generator illustrated. It can be used whenever it is desired to generate intermediate colors which are held in compressed form, provided that the compressed codes bear a linear relationship which parallels that of the actual colors themselves. In particular, it can be used in the original generation of the mip-maps held in the memory 22.

The embodiments of the invention which have been described and illustrated provide various improvements over the known systems. In particular, efficient use is made of memory bandwidth for trilinear filtering, especially when using texture compression caches. By generating the lower-level mip-map data from compressed upper-level mip-map data as it is required, accesses across page breaks are reduced. A quad cache arrangement can be used to guarantee one trilinear filtered pixel per clock period. The operation of the decompression logic is improved by using dynamic decompression resource allocation. Finally, the unification of the lower-level mip-map generation with the decompression logic provides a system with relatively low overhead.

- 24 -

It will be appreciated that many modifications may be made to the systems described and illustrate which represent only selected and presently preferred embodiments of the invention.